# How To Load Content Into A Databricks Workspace

This article describes a few methods you can use to load content into your Databricks workspace.

The content can consists of several types: - Notebooks - SQL Dashboards and its related queries - Data: tables and views - Delta Live Tables (DLT) Pipelines

The above list is illustrative, but not intended to be comprehensive

### Notebooks

Notebooks contain a mix of code, narrative and explanations that are a great way to encapsulate an experiment. Here are a couple of ways you can bring notebooks into your Databricks environment.

##### Method 1: Export/Import DBC Archives

1. Browse for [solution accelerators](https://www.databricks.com/solutions/accelerators) that contain solutions to common problems. Most solution accelerators come with notebooks in the form of .dbc archives
2. You can also create your own DBC archive by exporting an existing notebook or folder from your Databricks workspace. Simply choose File –> Export from the notebook menu, or Export from the workspace folder menu
3. Importing a DBC archive is equally simple. Just go to the workspace folder where you want to import and choose Import from the context menu

##### Method 2: Use Git Repos

Databricks supports git repositories from popular providers such as Github, Gitlab and Bitbucket. Create an empty repository on one of these services and load your notebooks into it. Or you may fork or clone an [existing repository](https://github.com/orgs/databricks/repositories) containing sample notebooks. Then, in your Databricks workspace, add the repository into the Repos section. You will need to configure Databricks with an access token from the service (eg. Github)

### SQL Dashboards and Queries

Databricks SQL allows you to create dashboards powered by SQL queries. Your Databricks environment provides a few samples in the [Dashboard Samples Gallery](https://docs.databricks.com/sql/get-started/sample-dashboards.html), and you can [create your own dashboards](https://docs.databricks.com/sql/user/dashboards/index.html) by combining queries and visualizations.

If you want to import dashboards and their underlying queries from another Databricks environment, you can use the dashboard migration tool [databricks-sql-clone](https://github.com/QuentinAmbard/databricks-sql-clone). It is recommended to add tags to identify the dashboards you want to migrate. Using this tool involves creating a config.json file with source and target credentials and specifying the tags you wish to migrate. See example below:

{  
 "source": {  
 "url": "https://e2-demo-field-eng.cloud.databricks.com",  
 "token": "dapi\_SOURCE\_TOKEN",   
 "dashboard\_tags": ["colab"]   
 },  
 "delete\_target\_dashboards": true,   
 "targets": [  
 {  
 "url": "https://enb-colab.cloud.databricks.com",  
 "token": "dapi\_TARGET\_TOKEN",  
 "endpoint\_id": "5381050c7c319b2d",   
 "permissions":[   
 {  
 "user\_name": "class+partners@databricks.com",  
 "permission\_level": "CAN\_MANAGE"  
 },  
 {  
 "group\_name": "users",  
 "permission\_level": "CAN\_RUN"  
 }  
 ]  
 }  
 ]  
}

### Data: tables and views

There are a few different ways to load data into your Databricks workspace: 1. Copy files into DBFS using the databricks CLI tool 2. [Mount a cloud storage bucket](https://docs.databricks.com/data/data-sources/aws/amazon-s3.html) (eg. S3) into DBFS 3. Set up [instance profiles](https://docs.databricks.com/administration-guide/cloud-configurations/aws/instance-profiles.html) to allow secure access to cloud object storage 4. Use temporary cloud credentials (eg. AWS session token) to copy files into the workspace

Methods 1 and 4 are described below. Refer to the documentation links for methods 2 and 3.

##### Method 1 - Dabricks CLI

Here is a sample script that uses the CLI to upload data to the DBFS FileStore

**copy\_tables.sh**

# Download files to local computer  
tables=("customer\_satisfaction" "products" "olist")  
for table in ${tables[@]}; do  
 echo $table  
 aws s3 cp --recursive s3://databricks-datasets-private/field-demos/retail/$table retail/$table  
done  
  
# Upload files to destination dbfs  
databricks fs cp --recursive retail dbfs:/FileStore/retail --profile colab

##### Method 4 - Temporary Cloud Credentials

This method is a bit hacky, but gets the job done if you run into permission issues with the earlier methods.

Assuming you have some data in your AWS account that you want to load into Databricks, first ensure that you are authenticated in the AWS CLI. You may have to run aws configure or gimme-aws-creds.

Then run the following command in your local Python interpreter to get your temporary credentials:

import boto3  
boto3.session.Session().get\_credentials().get\_frozen\_credentials()

Use the credentials obtained from the previous step in your Databricks notebook to set your cluster’s Hadoop configuration, as follows:

AccessKey = "ASIA\_YOUR\_ACCESS\_KEY"  
Secret = "YOUR\_SECRET\_KEY"  
Token = "YOUR\_SESSION\_TOKEN"  
sc.\_jsc.hadoopConfiguration().set("fs.s3a.aws.credentials.provider", "org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider")  
sc.\_jsc.hadoopConfiguration().set("fs.s3a.access.key", AccessKey )  
sc.\_jsc.hadoopConfiguration().set("fs.s3a.secret.key", Secret)  
sc.\_jsc.hadoopConfiguration().set("fs.s3a.session.token", Token)

See this [KB article](https://kb.databricks.com/en_US/notebooks/access-s3-temp-session-token) for more details

### Delta Live Tables (DLT) Pipelines